August 9, 2005

Statistical Analysis with R

Terry A. Cox, M.D., Ph.D.
National Eye Institute

Course Outline

1. The R website and documentation
2. Installing and updating R
3. The Windows GUI
4. R language essentials
5. R graphics
6. Basic statistics in R

URLs

- http://www.r-project.org/

  The source for R software and documentation. Links to ESS and R-Winedt, which provide R support for EMACS and Winedt, respectively.

- http://www.bioconductor.org/

  Bioconductor is an open source and open development software project for the analysis and comprehension of genomic data.

- http://www.cs.wisc.edu/∼ghost/

  Ghostscript and GSview, software for interpreting and viewing PostScript files.

- http://www.winedt.com/

  Shareware text editor for Windows designed for use with LaTeX. R-Winedt provides integration with R.

- http://www.textpad.com/

  An easy-to-use inexpensive text editor for Windows. Syntax definition files for R are available (see the add-ons directory at the website).

Documentation

R comes with several manuals in both html and pdf formats. Of particular relevance is *An Introduction to R*. Also the Contributed Documentation section at the R website contains several introductory manuals. In addition, the r-help mailing list is quite active, and search facilities are available for its archives. *R News*, available at the R website, is also very useful.

Books

- *Introductory Statistics with R*
  by Peter Dalgaard
  Publisher: Springer Verlag
  ISBN: 0387954759
  Publication Date: August 2002
  Paperback: 288 pages

  — Excellent for getting started with R. Covers basic statistical analysis, as well as linear models, logistic regression, and survival analysis.

- *Modern Applied Statistics with S*, 4th edition
  by Brian D. Ripley and William N. Venables
  Publisher: Springer Verlag
  ISBN: 0387954570
  Publication Date: July 2002
  Hardcover: 512 pages

  — Intermediate-level text that includes many state-of-the-art methods.

- *Regression Modeling Strategies*
  by Frank E. Harrell
  Publisher: Springer Verlag
  ISBN: 0387952322
  Publication Date: June 2001
  Hardcover: 582 pages

  — Lots of good stuff on linear models, logistic regression, and survival analysis.

- *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*
  by Robert Gentleman, Vincent Carey, Wolfgang Huber, Rafael Irizarry, Sandrine Dudoit
  Publisher: Springer
  ISBN: 0387251464
  Publication Date: August, 2005
  Hardcover: 464 pages

  — Written by developers of Bioconductor software.

The works by Verzani and Maindonald in the Contributed Documentation section at the R website have been published as books. An earlier version of the manual, *An Introduction to R*, has also been published as a book, as have some of the other manuals. These and other books on R can be found at amazon.com and bn.com.

R Examples

Preliminaries

Before proceeding install the packages **ISwR**, **car**, and **locfit**.

If you want to type in the following listings, replace the left arrow symbols with “<–”. The equals sign, “=”, can also be used in recent versions of R. All listings are available in the file, Examples.R.

Vectors

```
x ← c(92,63,22,32,56,80,51,14,21,38) # Or x ← scan()
x
x[1]
x[2:4]
x[seq(1,9,2)]
?seq
x[-1]
x[-(2:4)]
x[c(1,3,5)]
(x > 50) # Put a space between < and negative numbers!
x[x > 50]
which(x > 50)
sort(x)
rev(x)
c.x ← c(rep("Boy",5),rep("Girl",5))
# Or c.x ← rep(c("Boy","Girl"), c(5,5))
is.character(c.x)
mode(c.x)
```

The name of an object must start with a letter (A–Z and a–z) and can include letters, digits (0–9), and dots (.). R is case-sensitive, so that x and X can name two distinct objects.

Computations

```
2*x
x^2
xbar ← mean(x)
xv ← (x - xbar)^2
xvar ← sum(xv)/(length(x)-1)
xsd ← sqrt(xvar)
sd(x)
```

Missing Data

```
xm ← x
xm[x>50] ← NA
xm
mean(xm)
mean(xm, na.rm=TRUE)
```

Matrices

```
y ← c(79,24,38,45,64,58,20,53,15,83)

z ← cbind(x,y)
z[,"x"]
colnames(z)
which(z>50)
which(z>50, arr.ind=T)
matrix(y, nrow=2, byrow=TRUE)
matrix(1,2,3)
zcp ← t(z) %*% z
diag(zcp)
diag(3)
```

Lists and data frames

```
zLst ← list(first=x, second=y, gender=c.x)
zLst$first

zDf ← data.frame(first=x, second=y, gender=c.x)
zDf$first
zDf[zDf$first>50,]

lapply(zDf[,1:2], mean)
sapply(zDf[,1:2], mean) # See also apply, mapply, tapply
lapply(zDf[,sapply(zDf,is.numeric)], mean)

zDf[grep("b", as.character(zDf$gender), ignore.case=TRUE),]
```

Miscellaneous

```
search()
ls()
rm(zLst)
?"%*%"
help.search("fisher")
options()

library(ISwR)
data(kfm)
?kfm
```

The function, source(), can be used to input R code from an external file. R objects such as dataframes can be saved using the save() function, and read into another R session using the load() function. See also the function, sink().

Data entry

Use / or \\, not \, in path names.

```
kDf ← read.delim("H:/R Course/kfm.txt") # Use your own filename and path
summary(kDf)
str(kDf)
```

```
attach(kDf)
search()
mat.height
detach(kDf)
```

See also the R manual, *R Data Import/Export*, and the package, **foreign**.

Summary plots

```
hist(kDf$weight, xlab="Weight (kg)")

boxplot(weight ∼ sex, data=kDf, boxwex=0.3, ylab="Weight (kg)", names=c("Boys","Girls"))
```

See Figure 1 for an example of a dot plot, an alternative to bar graphs.

Scatter plots

```
plot(kDf$mat.weight, kDf$weight)
plot(weight ∼ mat.weight, data=kDf)

attach(kDf)
library(locfit)
fit ← locfit(weight ∼ mat.weight)
plot(fit, band="global")
points(mat.weight, weight, pch=20, col="gray50")
detach(kDf)

library(car)
scatterplot(weight ∼ mat.weight, reg.line=lm, smooth=TRUE, labels=FALSE,
    boxplots='xy', span =0.5, data=kDf)

pairs(kDf[,c("weight","mat.weight","mat.height")])
```

Cumulative histogram

The following code plots the empirical cumulative distribution function:

```
attach(kDf)
boy.wt ← weight[sex=="boy"]
girl.wt ← weight[sex=="girl"]
m ← length(boy.wt)
n ← length(girl.wt)
plot(sort(boy.wt), (1:m)/m, type="s", ylim=c(0,1), xlim=range(weight),
    xlab="Weight", ylab="Cumulative frequency", lty=1)
lines(sort(girl.wt), (1:n)/n, type="s", lty=2)
legend(c(6,6.5), c(0.14,0.3), legend=c("Boys","Girls"), lty=1:2)
detach(kDf)
```

See also the function, ecdf(), in Frank Harrell's **Hmisc** library.

Line plots

```
x ← seq(-4,4,0.01)
y ← dnorm(x)
plot(x, y, type="l", main="Normal Density", xlab="x", ylab=substitute(paste(phi, "(x)")))
text(-3, 0.2, expression(phi), cex=2, col="gold") # For fun
phi ← (sqrt(5) + 1)/2
text(3, 0.2, phi)
```

Plot output

The following code produces Figure 2.

```
attach(kDf)
postscript(file="H:/R Course/Fig2.ps", horizontal=F, width=5, height=5)
# Use your own filename and path
plot(weight ~ mat.weight, type="n", xlab="Maternal weight", ylab="Infant weight")
points(weight[sex=="boy"] ~ mat.weight[sex=="boy"], pch=19, col="blue")
points(weight[sex=="girl"] ~ mat.weight[sex=="girl"], pch=19, col="red")
dev.off()
detach(kDf)
```

Postscript files produce publication-quality graphics on laser printers, and they can be used in LaTeX documents. To create an encapsulated postscript file that can be imported into MS Word, substitute the following line:

```
postscript(file="H:/R Course/Fig2.eps", horizontal=FALSE, onefile=FALSE,
    paper="special", width=5, height=5)
```

The following code produces a graphic that can be imported into MS Powerpoint:

```
win.metafile("H:/R Course/boxplot.wmf")
# Use your own filename and path
old.par ← par(no.readonly=TRUE)
line.col ← "gray"
par(fg=line.col, col.axis=line.col, col.lab=line.col, cex=1.5, lwd=2)
boxplot(weight ~ sex, data=kDf, boxwex=0.3, ylab="", names=c("Boys","Girls"),
    notch=TRUE, col=heat.colors(2), border=line.col)
mtext("Weight (kg)", side=2, line=2.5, cex=1.5)
par(old.par)
dev.off()
```

Summary statistics

```
mean(kDf$weight)
sd(kDf$weight)
quantile(kDf$weight)
median(kDf$weight)
IQR(kDf$weight)
mad(kDf$weight)

table(kDf$sex)
```

Tabular data

```
wtable ← table(kDf$sex, (kDf$weight < 5.5) )
ft ← fisher.test(wtable)
ct ← chisq.test(wtable, correct=FALSE)
```

For other relevant functions, see the documentation for the package, **ctest**. The package, **vcd**, contains functions for Cohen's kappa and weighted kappa, among others.

t tests

```
t.test(weight ~ sex, data=kDf)
wilcox.test(weight ~ sex, data=kDf)
```

Correlation

```
cor(kDf[,c("weight","mat.weight","mat.height")])
cor(kDf[,c("weight","mat.weight","mat.height")], method="spearman")
```

See also the function, cor.test(), in the package, **ctest**.

Linear regression

```
kLm ← lm(weight ~ mat.weight, data=kDf)
summary(kLm)
kLm.summ ← summary(kLm)
names(kLm)
names(summary(kLm))
summary(kLm)$r.squared
kLm$coefficients
summary(kLm)$coefficients
plot(kLm)

# Regression lines and confidence intervals

pwt ← seq(min(kDf$mat.weight), max(kDf$mat.weight), 0.05)
clim ← predict(kLm, data.frame(mat.weight=pwt), interval="c")
plim ← predict(kLm, data.frame(mat.weight=pwt), interval="p")
plot(weight ~ mat.weight, data=kDf, ylim=range(plim[,2:3]))
lines(pwt, clim[,1], lty=1, col="black")
lines(pwt, clim[,2], lty=2)
lines(pwt, clim[,3], lty=2)
lines(pwt, plim[,2], lty=3)
lines(pwt, plim[,3], lty=3)

# Alternatively, add lines as follows:

matlines(pwt, clim, lty=c(1,2,2), col="black")
matlines(pwt, plim[,2:3], lty=3, col="black")
```

Logistic regression

```
kGlm ← glm(sex ~ mat.weight*mat.height, data=kDf, family="binomial")
summary(kGlm)
```

Function creation

```
summVar ← function(y) {
    qy ← quantile(y, na.rm=T)
    names(qy) ← c("minimum","1st quartile","median","3rd quartile","maximum")
    c(N = length(y),
      mean = mean(y, na.rm=T),
      "st. dev." = sd(y, na.rm=T),
      qy[1], qy[2], qy[3], qy[4], qy[5],
      IQR = IQR(y, na.rm=T),
      "mean abs. dev." = mad(y, na.rm=T),
      missing = sum(is.na(y)),
      "Shapiro-Wilk test" = shapiro.test(y)$p.value )
}

procUnivariate ← function(y) {
    if (is.numeric(y)) {
        out ← summVar(y)
    }
    else {
        ynum ← y[,sapply(y,is.numeric)]
        if (is.list(ynum)) out ← sapply(ynum, summVar) # or lapply
        else {
            out ← list(summVar(ynum))
            names(out) ← names(y)[sapply(y,is.numeric)]
        }
    }
    return(out)
}

options(digits=4)
procUnivariate(kDf)

procUnivariate
```

See also the function, describe(), in Frank Harrell's **Hmisc** library.

Random numbers

```
sample(1:10)
sample(LETTERS, size=10, replace=T)
set.seed(10); runif(1)

kDf[sample(1:length(kDf$no), size=5),]

x ← rnorm(1000)
hist(x,nclass=20)
```

Functions are automatically available for a number of distributions in R. See also the packages, **bindata**, **mvtnorm**, **SuppDists**, **MCMCpack**, and **MASS**.

```
# Poker

suit ← rep(c("Diamonds","Clubs","Spades","Hearts"), rep(13,4))
card ← rep(c(2:10,"Jack","Queen","King","Ace"), 4)
deck ← paste(card, suit, sep=" of ")
```

```
shuffle ← sample(deck)
player.1 ← shuffle[1:5]
player.2 ← shuffle[6:10]
```

Create external dataset

```
kDf$wtbin ← (kDf$weight < 5.5)

zz ← file("H:/R Course/temp.data", "w")
write.table(kDf, file=zz, sep="\t", quote=FALSE, row.names=FALSE)
close(zz)
```

Upgrading R

From the R for Windows FAQ:

> 2.5 How do I UNinstall R?
>
> Normally you can do this from the R group on the Start Menu or from the Add/Remove Programs in the Control Panel. If it does not appear there or if you want to remove an old version, run unins000.exe in the top-level installation directory. (There should be a separate uninstall item in the R group for each installed version of R.)
>
> Uninstalling R only removes files from the initial installation, not (for example) packages you have installed.
>
> If all else fails, you can just delete the whole directory in which R was installed.
>
> 2.6 What's the best way to upgrade?
>
> That's a matter of taste. For most people the best thing to do is to uninstall R (see the previous Q), install the new version, copy any installed packages to the library folder in the new installation, run update.packages() in the new R ('Update packages from CRAN' from the Packages menu, if you prefer) and then delete anything left of the old installation. Different versions of R are quite deliberately installed in parallel folders so you can keep old versions around if you wish.
>
> Upgrading from R 1.x.y to R 2.0.0 is special as all the packages need to be reinstalled. Rather than copy them across, make a note of their names and re-install them from CRAN.

Upgrading in Linux and Mac OS X is simpler: just install the new version over the old version.

`q()`

tac@nei.nih.gov

Figure 1: Example of a dot plot.

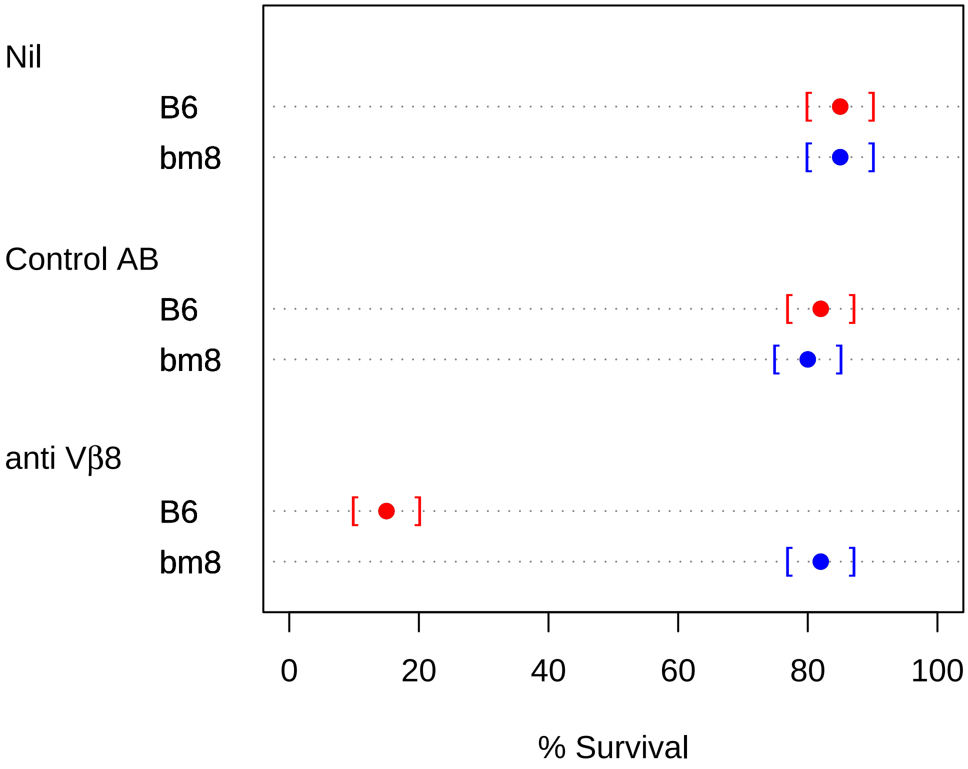
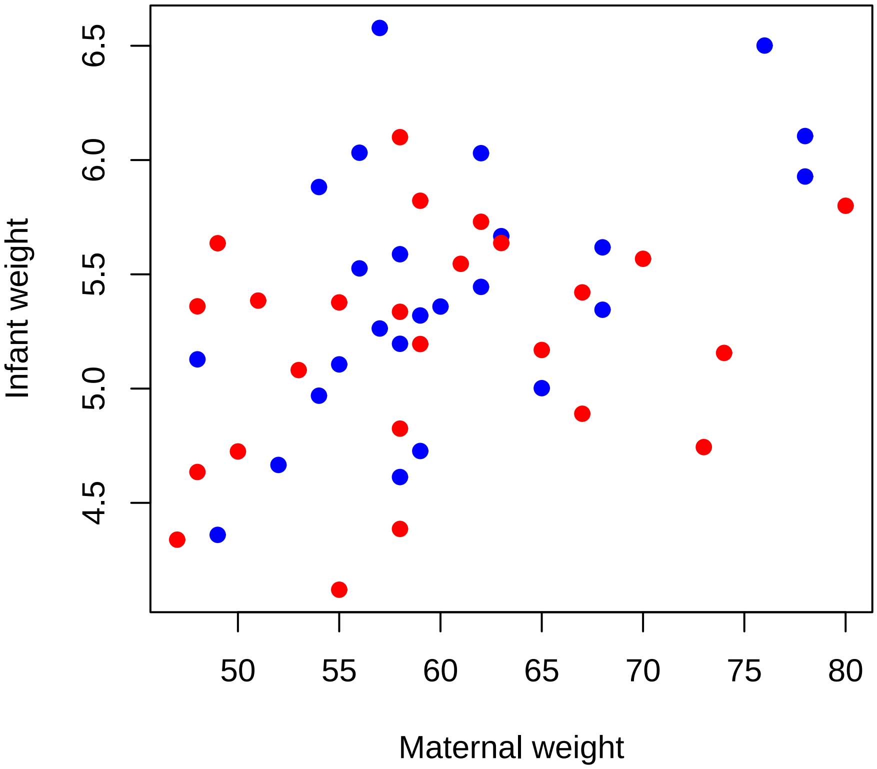


Figure 2: Example of a plot good enough to publish.



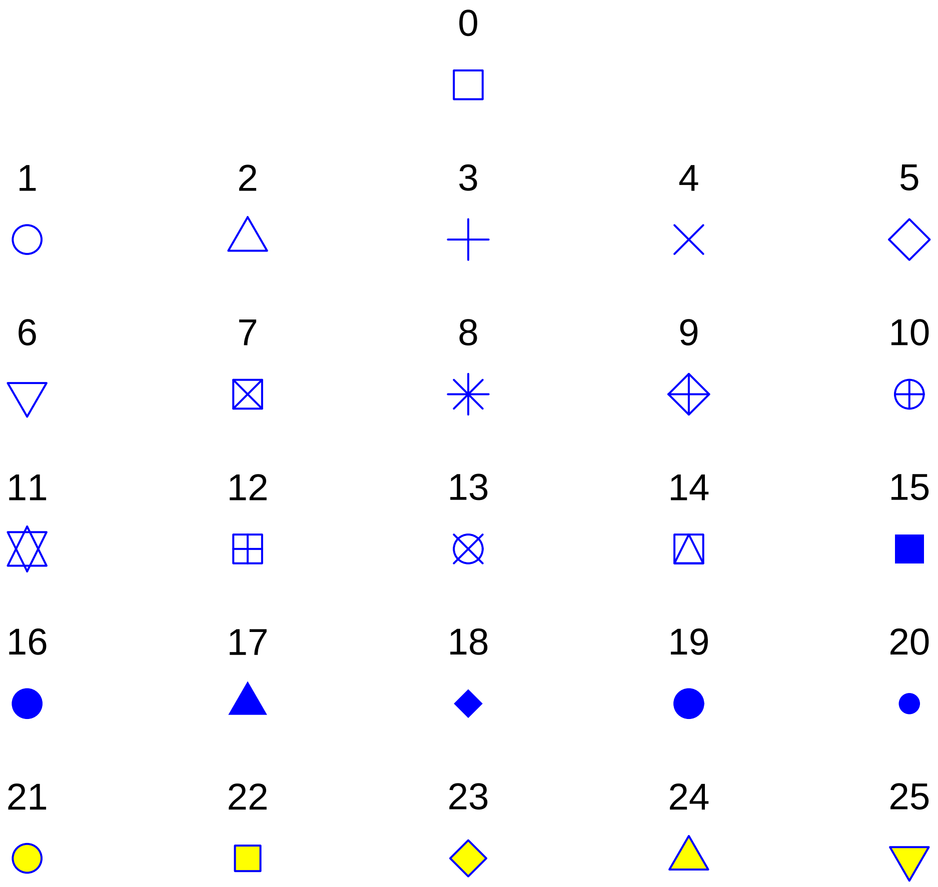


Figure 3: R plotting symbols.

Figure 3 was produced using the following code:

```
postscript(file="H:/R Course/chars.ps", horizontal=F, width=5, height=5)
op ← par(no.readonly=TRUE)
par(mar=rep(0.1,4))
x ← c(3,rep(1:5,5))
y ← c(6,rep(5:1,rep(5,5)))
z ← y + 0.4
plot(x, y, pch=0:25, type="n", axes=FALSE, xlab="", ylab="", ylim=c(1,6.7))
points(x, y, pch=0:25, cex=2, col="blue", bg="yellow")
text(x, z, labels=as.character(0:25), cex=1.2)
par(op)
dev.off()
```